

AD-A037 616

CARNEGIE-MELLON UNIV PITTSBURGH PA MANAGEMENT SCIENC--ETC F/G 12/1
SET COVERING WITH CUTTING PLANES FROM CONDITIONAL BOUNDS.(U)
JAN 77 E BALAS

N00014-75-C-0621

UNCLASSIFIED

RR-399

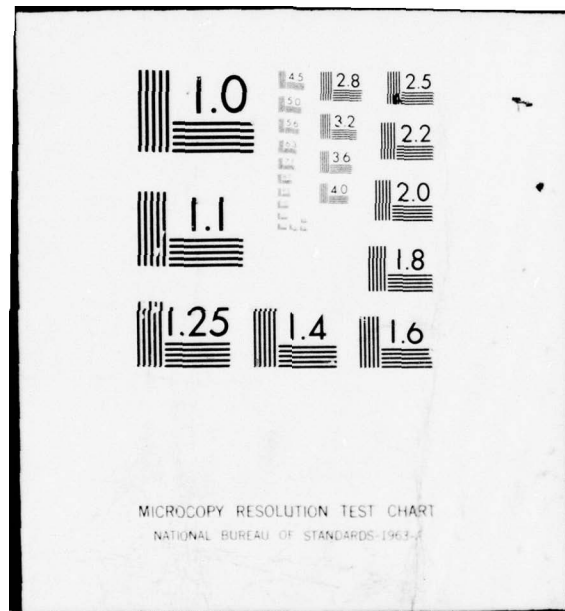
NL

| OF |
AD
A037616

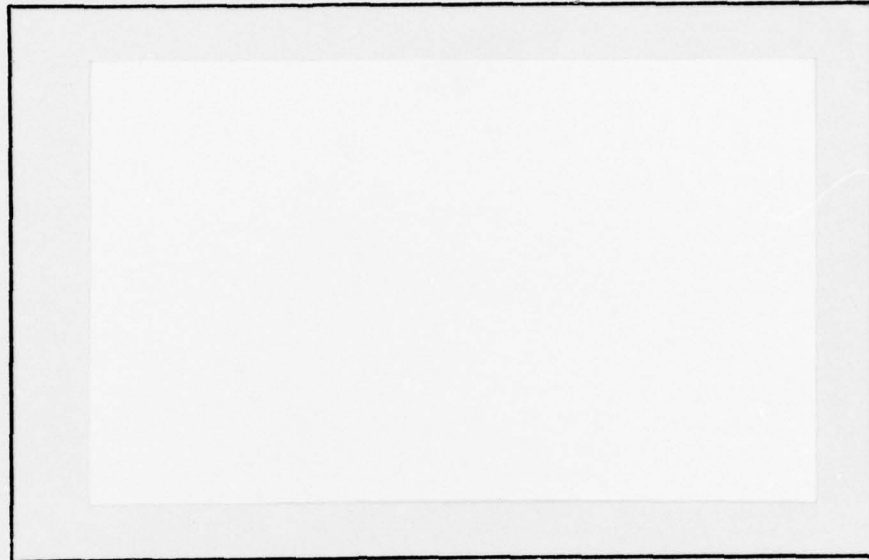


END

DATE
FILMED
4-77



ADA037616



1
B.S.

See 1473
in back

Carnegie-Mellon University

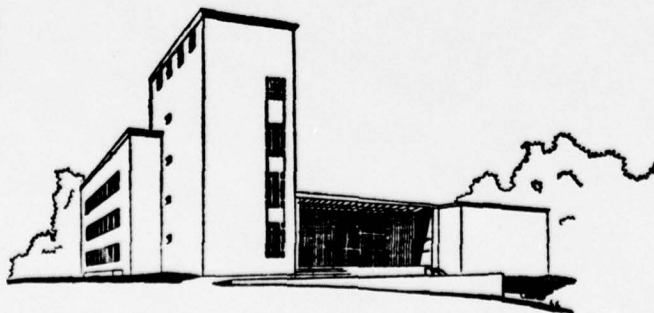
PITTSBURGH, PENNSYLVANIA 15213

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

AD No. _____
DDC FILE COPY

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



DDC
RECEIVED
APR 1 1977
A

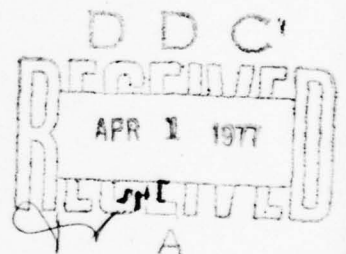
DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Management Sciences Research Report No. 399

SET COVERING
WITH CUTTING PLANES FROM
CONDITIONAL BOUNDS*

by
Egon Balas

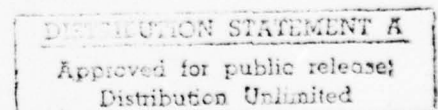
July 1976
Revised January 1977



*This paper was presented at the 9th International Symposium on
Mathematical Programming, Budapest, August 23-27, 1976

This report was prepared as part of the activities of the Management
Sciences Research Group, Carnegie-Mellon University, under Grant
MPS 73-08534 A02 of the National Science Foundation and Contract
N00014-75-C-0621 NR 047-048 with the U.S. Office of Naval Research.
Reproduction in whole or in part is permitted for any purpose of the
U.S. Government.

Management Science Research Group
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213



ABSTRACT

A conditional lower bound on the minimand of an integer program is a number which would be a valid lower bound if the constraint set were amended by certain inequalities, also called conditional. If such a conditional lower bound exceeds some known upper bound, then every solution better than the one corresponding to the upper bound violates at least one of the conditional inequalities. This yields a valid disjunction, which can be used to partition the feasible set, or to derive a family of valid cutting planes. In the case of a set covering problem, these cutting planes are themselves of the set covering type. The family of valid inequalities derived from conditional bounds subsumes as a special case the Bellmore-Ratliff inequalities generated via involutory bases, but is richer than the latter class and contains considerably stronger members, where strength is measured by the number of positive coefficients. The paper discusses two algorithms based on cutting planes from conditional bounds. None of them uses the simplex method (though a variant based on the latter is also feasible). Some computational experience is presented.

ACCESSION for	
NTIS	Write Section <input checked="" type="checkbox"/>
DDC	Putt Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION AVAILABILITY CODES	
DISC.	AVAIL. AND OR SPECIAL
A	

Contents

	<u>Page</u>
1. Introduction.....	1
2. Conditional Bounds.....	2
3. Cutting Planes From Conditional Bounds.....	10
4. Some Properties of Cuts from Conditional Bounds.....	17
5. Generating Cuts.....	23
6. Generating Prime Covers.....	28
7. Generating Feasible Solutions to (D).....	33
8. Algorithms.....	37
9. Numerical Example.....	40
10. Preliminary Computational Experience.....	45
References.....	48

SET COVERING WITH CUTTING PLANES
FROM CONDITIONAL BOUNDS

by

Egon Balas

1. Introduction

We consider the set covering problem

$$(SC) \quad \min \{cx \mid Ax \geq e, \ x_j = 0 \text{ or } 1, \ j \in N\}$$

where $A = (a_{ij})$ is $m \times n$, $e \in R^m$, $e = (1, \dots, 1)$, $c \in R^n$, and $a_{ij} \in \{0, 1\}$, $i \in M = \{1, \dots, m\}$, $j \in N = \{1, \dots, n\}$. We will denote by a^i and a_j the i -th row and j -th column of A , respectively. Without loss of generality, we assume that $c_j > 0$, $\forall j \in N$. Using established terminology, we call a vector x satisfying the constraints of (SC) a cover, and the set of indices j such that $x_j = 1$, the support of the cover. A cover is called prime if no proper subset of its support defines a cover.

This problem, and its equality-constrained counterpart, the set partitioning problem, are useful mathematical models for a great variety of scheduling and other important real world problems, like crew scheduling, truck delivery, tanker routing, information retrieval, fault detection, stock cutting, offshore drilling platform location, etc., and a literature of considerable size exists on solution methods for these models (see [6] for a survey of set covering and set partitioning; [5] for a computational study and comparison of several solution techniques; [3], [8] and [9] for some of these methods; and [2] for a more recent survey of set partitioning, which also contains a bibliography of applications of both models).

In this paper we propose a new approach to set covering, based on the idea of conditional bounds. In section 2 we introduce this concept for arbitrary mixed integer programs, and show how it can be used to derive valid disjunctions. The latter in turn can be used either to partition the feasible set in the framework of a branch and bound approach, or to derive a family of valid cutting planes. In the case of a set covering problem, the cutting planes derived from conditional bounds are themselves of the set covering type. These cuts are discussed in section 3, where the Bellmore-Ratliff inequalities [3] generated via involutory bases are shown to be a special case of the larger family of cutting planes defined in this paper. In section 4 we examine the conditions under which a cutting plane derived from a conditional bound cuts off a specified prime cover. Since the family of cuts from conditional bounds is too large to be generated in its entirety, in section 5 we discuss a procedure for generating "strong" members of the family. Sections 6 and 7 discuss heuristics for generating "good" prime covers and feasible solutions to the dual of the linear program associated with (SC), which are needed to generate convenient cuts. Next we state two algorithms based on cutting planes from conditional bounds (section 8). Section 9 contains a numerical example, and in section 10 we discuss some early computational experience.

2. Conditional Bounds

The central idea of our approach is to derive valid inequalities for the set covering problem from conditional bounds. Since this concept is meaningful for arbitrary mixed integer programs, we will introduce it in this more general context.

A conditional lower bound on the (objective function) value of a mixed integer program (P) is a number which is a valid lower bound if (P) is amended by some inequalities. The inequalities used to derive the conditional bound are called conditional. The purpose of these inequalities is to produce a conditional lower bound at least equal to a known upper bound. If this is achieved, then at least one of the conditional inequalities is violated by any solution better than the one associated with the upper bound, and this yields a valid constraint.

To be more specific, consider the integer program

$$(P) \quad \min \{cx \mid Ax \geq b, \ x \geq 0, \ x_j \text{ integer, } j \in N\},$$

where A is an arbitrary $m \times n$ matrix ($n = |N|$) and b is an arbitrary m -vector. The pair of dual linear programs associated with (P) is

$$(L) \quad \min \{cx \mid Ax \geq b, \ x \geq 0\}$$

and

$$(D) \quad \max \{ub \mid uA \leq c, \ u \geq 0\}.$$

For any feasible optimization problem (S), let $z(S)$ be the value of (an optimal solution to) (S).

Any feasible solution x to (P) provides an upper bound cx on $z(P)$, and any feasible solution u to (D) provides a lower bound ub on $z(D) = z(L)$, hence also on $\overline{z(P)}$.

Let u be a feasible solution to (D), i.e., such that

$$(1) \quad uA \leq c, \quad u \geq 0,$$

and suppose the constraints of (L) and (P) are amended with the set of (conditional) inequalities

$$(2) \quad Cx \geq e_p,$$

where $C = (c_{ij})$ is a $p \times n$ matrix of 0's and 1's, and e_p is the p -vector of 1's, with $1 \leq p \leq n$. Suppose also that C has no zero rows, i.e.,

$$(3) \quad \sum_{j=1}^n c_{ij} \geq 1, \quad i = 1, \dots, p.$$

Further, denote by (P_C) and (L_C) the problems obtained by amending the constraint sets of (P) and (L), respectively, with the set (2) of conditional inequalities, and let (D_C) be the dual of the linear program (L_C) .

Let u be a m -vector satisfying (1). If there exists a p -vector v , $v \geq 0$, $v \neq 0$, such that

$$(4) \quad vC \leq c - uA,$$

then (u, v) is a feasible solution to (D_C) and therefore $ub + ve_p$ is a lower bound on $z(D_C) = z(L_C)$, hence also on $z(P_C)$. We will say in this case that $ub + ve_p$ is a conditional lower bound on $z(P)$.

Now let z_U be a known upper bound on $z(P)$. If

$$(5) \quad ve_p \geq z_U - ub,$$

i.e., if the conditional lower bound on $z(P)$ exceeds or equals the upper bound z_U , then

$$\begin{aligned} z(P_C) &\geq z(L_C) \\ &\geq ub + ve_p \geq z_U, \end{aligned}$$

and hence every feasible solution to (P) better than the one associated with the bound z_U violates at least one of the inequalities (2), i.e., satisfies the disjunction

$$\bigvee_{i=1}^p \left(\sum_{j=1}^n c_{ij} x_j \geq 1 \right).$$

Since C is a 0-1 matrix and x is a 0-1 vector, this disjunction is the same as

$$\bigvee_{i=1}^p \left(\sum_{j=1}^n c_{ij} x_j = 0 \right).$$

If we denote

$$Q_i = \{j \in N \mid c_{ij} = 1\}, \quad i = 1, \dots, p,$$

the result that we have just proved can be stated as follows.

Theorem 2.1. Let z_U be a known upper bound on $z(P)$, and let u satisfy (1). If there exists a $p \times n$ matrix $C = (c_{ij})$, $c_{ij} \in \{0, 1\}$, $\forall i, j$ ($1 \leq p \leq n$), satisfying (3), and a p -vector $v \geq 0$, $v \neq 0$, satisfying (4) and (5), then every feasible solution x to (P) such that $cx < z_U$ satisfies the disjunction

$$(6) \quad \bigvee_{i=1}^p (x_j = 0, j \in Q_i).$$

We have stated Theorem 2.1 for the pure integer case in order to simplify the exposition, but it (and the rest of this section) is easily seen to carry over to the case of a mixed integer program. Indeed, if $N_1 \subset N$ is the set of integer-constrained variables, $N_1 \neq N$, let $c_{ij} = 0$, $\forall i = 1, \dots, p$, $\forall j \in N \setminus N_1$, and the results are valid with some changes in the notation.

Note that the disjunction (6), though derived from a conditional bound, is an "unconditionally" valid constraint.

The first question that arises in connection with Theorem 2.1, is that of the existence of a matrix C and a vector v satisfying the above requirements.

Theorem 2.2. Let z_U be a known upper bound on $z(P)$, let u be a feasible solution to (D), and let

$$(7) \quad s = c - uA.$$

Then there exists a pair C, v satisfying the requirements of Theorem 2.1 if and only if

$$(8) \quad \sum_{j \in N} s_j \geq z_U - ub.$$

Proof. Let u and s satisfy (1) and (7). If (8) holds, then the pair C, v , defined by $C = I$ (the identity matrix of order N) and $v_j = s_j, j \in N$, satisfies (3), (4) and (5).

Conversely, if C and v satisfy (3), (4) and (5), then adding the inequalities (4) and substituting s for $c - uA$ yields

$$\sum_{j \in N} s_j \geq \sum_{i=1}^p v_i \left(\sum_{j \in N} c_{ij} \right)$$

$$\geq v_e \quad [\text{from (3)}]$$

$$\geq z_U - ub \quad [\text{from (5)}] \quad \text{Q.E.D.}$$

Note that, if $p = 1$, i.e., if C has a single row, then the disjunction (6) becomes $x_j = 0, j \in Q_1$. Somewhat more generally, we have the following.

Remark 2.1. Let z_U, u and s be as in Theorem 2.2, and define

$$Q_0 = \{j \in N | s_j \geq z_U - ub\}.$$

Then every feasible solution x to (P) such that $cx < z_U$ satisfies

$$x_j = 0, \quad j \in Q_0.$$

Thus, whenever $Q_0 \neq \emptyset$, the variables indexed by Q_0 can be set to zero permanently.

Example 1. Let (P) be an integer program (minimization problem) with 10 variables, and let $z_U = 35$ be the value of some known integer solution. Further, let u be a feasible solution to the dual of the associated linear program, with $ub = 27.9$, and let the reduced costs s_j associated with ub be

j	1	2	3	4	5	6	7	8	9	10
s_j	1.5	2.4	0	3.1	0	5.8	3.3	2.6	3.2	4.7

We have $z_U - ub = 35 - 27.9 = 7.1$. To construct a pair C, v satisfying the conditions of Theorem 2.1, we first choose $v_1 = s_1, v_2 = s_4$ and $v_3 = s_8$; then $v_1 + v_2 + v_3 = 7.2 \geq 7.1$. Next we choose the 1's among the elements c_{ij} of C in such a way that $v_1 c_{1j} + v_2 c_{2j} + v_3 c_{3j} \leq s_j$ for $j = 1, \dots, 10$. (Neither v , nor C is of course unique.) C and v are shown below, where the blanks are 0's.

	1	2	3	4	5	6	7	8	9	10	v
$C =$	1	1								1	1.5
				1		1	1		1		3.1
						1		1		1	2.6

From Theorem 2.1, every x satisfying the constraints of (P) and such that $cx < 35$ satisfies the disjunction

$$x_1 = x_2 = x_{10} = 0 \quad \vee \quad x_4 = x_6 = x_7 = x_9 = 0 \quad \vee \quad x_6 = x_8 = x_{10} = 0.$$

Given a feasible solution u to (D) whose associated reduced costs s satisfy (8), it is usually not difficult to find a pair C, v satisfying the conditions of Theorem 2.1; but the problem is to find one which defines a conveniently strong disjunction (6). The following procedure provides a general framework for doing this, which allows for many variants.

1. Choose a minimum-cardinality subset S of N , such that

$$(9) \quad \sum_{j \in S} s_j \geq z_U - ub,$$

and order $S = \{j(1), \dots, j(p)\}$ according to decreasing values of $s_{j(i)}$. Then set $v_i = s_{j(i)}$, $i = 1, \dots, p$, and $c_{hj(i)} = 1$ for $h = i$, $c_{hj(i)} = 0$ for $h \neq i$, $i = 1, \dots, p$.

2. For $j \in N \setminus S$, define recursively for $i = 1, \dots, p$,

$$c_{ij} = \begin{cases} 0 \text{ or } 1 & \text{if } \sum_{h=1}^{i-1} v_h c_{hj} + v_i \leq s_j \\ 0 & \text{otherwise.} \end{cases}$$

The option of setting $c_{ij} = 1$ or $c_{ij} = 0$ in step 2 represents a choice between including i into Q_i , or leaving it available for inclusion into one or more sets Q_{i+k} . This can be decided by efficiency criteria, as will be seen later. It is easy to check that any pair C, v constructed by this procedure satisfies the requirements of Theorem 2.1.

A disjunction of the form (6) obtained from a conditional bound can be used to partition the feasible set in the framework of a branch and bound procedure, by creating p subproblems defined by the constraints

$$x_j = 0, \quad j \in Q_1$$

$$\sum_{j \in Q_1} x_j \geq 1; \quad x_j = 0, \quad j \in Q_2$$

.....

$$\sum_{j \in Q_1} x_j \geq 1, \quad \dots, \quad \sum_{j \in Q_{p-1}} x_j \geq 1; \quad x_j = 0, \quad j \in Q_p$$

For certain classes of integer programs, this way of branching seems highly efficient. It is currently being tested, for instance, in a new penalty method for solving traveling salesman problems [1], with excellent computational results.

Another way of using the disjunction (6) is to generate from it a family of valid inequalities. In the case of a set covering problem, these inequalities turn out to be of the set covering type, as shown in the next section.

In a broader context, the idea of deriving a valid ("unconditional") constraint from one or several conditional constraints may have many other applications. One of them appears in [7], where a properly chosen inequality is used to derive a bound from the fact that either the inequality or its complement must be satisfied by any feasible solution.

3. Cutting Planes From Conditional Bounds

From now on, we address ourselves to the set covering problem (SC) introduced in section 1; i.e., $A = (a_{ij})$ with $a_{ij} \in \{0,1\}$, $\forall i,j$, and $b = e$, where e is the m -vector of 1's. We will denote

$$N_i = \{j \in N \mid a_{ij} = 1\}, j \in M.$$

Theorem 3.1. Suppose the conditions of Theorem 2.1 are satisfied, i.e., the disjunction (b) is a valid constraint for (SC). With each $i \in \{1, \dots, p\}$, associate an index $h(i) \in M$, such that $N_{h(i)} \cap Q_i \neq \emptyset$. Then denoting

$$W = \bigcup_{i=1}^p [N_{h(i)} \setminus Q_i],$$

every cover x such that $cx < z_U$ satisfies

$$(10) \quad \sum_{j \in W} x_j \geq 1.$$

Proof. For $i = 1, \dots, p$, the i -th term of the disjunction (6) implies

$$\sum_{j \in N_h \setminus Q_i} x_j \geq 1, \quad \forall h \in M.$$

Hence, for any choice of indices $h(i) \in M$, $i = 1, \dots, p$, (6) implies the disjunction

$$\bigvee_{i=1}^p \left(\sum_{j \in N_{h(i)} \setminus Q_i} x_j \geq 1 \right),$$

which in turn implies (10).

Q.E.D.

The cutting planes of Theorem 3.1 are set covering inequalities, valid in the sense of being satisfied by every cover better than a given one. Since these properties are the same as those of the Bellmore-Ratliff cuts [3] obtained by the use of involutory bases, we next examine the relationship between the latter and our inequalities from conditional bounds. First, we show in the next theorem that the Bellmore-Ratliff inequalities are a subclass of the class of inequalities defined by Theorem 3.1. Then we show by way of example that the subclass in question is a proper one.

Theorem 3.2. The Bellmore-Ratliff inequalities [3] are a subclass of the class (10).

Proof. Let \bar{x} be a prime cover, B an involutory basis associated with \bar{x} , and $c_j = c_B a_j$ the j -th reduced cost, where c_B is the m -vector whose i -th component is $c_{j(i)}$, if the basic variable associated with row i is (the structural variable) $x_{j(i)}$, and 0 if the basic variable

associated with row i is a slack. (When B is an involutory basis, the reduced costs are known to be of the above form.) The Bellmore-Ratliff cut associated with \bar{x} and B is then

$$(11) \quad \sum_{j \in F} x_j \geq 1$$

where

$$F = \{j \in N \mid c_j - c_B a_j < 0\}.$$

To obtain this cut from a conditional bound, let $I_1 = \{j(1), \dots, j(p)\}$ be the index set of the structural basic variables, and set $u = 0$, $s = c$, and $v_i = c_{j(i)}$, $j(i) \in I_1$. Then u satisfies (1) and v satisfies (5) (with equality) for $z_U = \bar{c}x$.

We now construct the matrix $C = (c_{ij})$ of Theorem 2.1 as follows. Let $h(i)$ be the row index associated with basic variable $x_{j(i)}$. Define

$$c_{ij} = \begin{cases} a_{h(i),j} & , \quad j \in N \setminus F \\ 0 & , \quad j \in F \end{cases} \quad i = 1, \dots, p;$$

then

$$Q_i = N_{h(i)} \setminus F, \quad i = 1, \dots, p.$$

C trivially satisfies (3). To see that together with v it also satisfies (4), note that for $j \in N \setminus F$,

$$\begin{aligned} \sum_{i=1}^p v_i c_{ij} &= \sum_{i=1}^p c_{j(i)} a_{h(i),j} \\ &\leq c_j - u a_j; \end{aligned}$$

where the inequality follows from the fact that $c_j - \sum_{i=1}^p c_{j(i)} a_{h(i),j}$ is the j -th reduced cost (nonnegative for $j \in N \setminus F$), while $u = 0$. Further, for $j \in F$,

$$\begin{aligned} \sum_{i=1}^p v_i c_{ij} &= 0 \\ &\leq c_j - u a_j. \end{aligned}$$

Thus C and v satisfy the conditions of Theorem 2.1. Applying Theorem 3.1, we now associate with each $i \in \{1, \dots, p\}$ the row index $h(i)$, and define

$$\begin{aligned} W &= \bigcup_{i=1}^p [N_{h(i)} \setminus Q_i] \\ &= \bigcup_{i=1}^p [N_{h(i)} \cap F] \quad (\text{from the definition of the sets } Q_i) \\ &= \left[\bigcup_{i=1}^p N_{h(i)} \right] \cap F. \end{aligned}$$

On the other hand, from the definition of F , $j \in F$ implies $a_{h(i),j} = 1$ for some $i \in \{1, \dots, p\}$, hence

$$F \subseteq \left[\bigcup_{i=1}^p N_{h(i)} \right].$$

Thus $W = F$, and the cut (10) of Theorem 3.1 is in this case identical with (11). Q.E.D.

Thus the Bellmore-Ratliff cuts are a special case of the cuts (10). Furthermore, they are a proper subclass of the class (10), i.e., the conditional bound approach yields other inequalities besides those obtainable via involutory bases. Some of those other inequalities are considerably stronger, in the sense of having fewer positive coefficients. This is illustrated by the next example.

Example 2. Consider the set covering problem whose costs c_j and coefficient matrix A are shown in Tableau 1.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
c_j	3	1	1	3	1	2	2	3	3	3	3	3	3	4	4	4	5	6	8	9
1						1							1				1	1		1
2	1								1						1	1	1	1		1
3		1						1		1									1	
4			1											1						1
5				1					1	1							1		1	
6					1					1				1		1				1
7							1								1		1		1	1
8												1	1						1	
9								1						1			1	1		1
10										1			1			1			1	
11						1						1						1	1	1

Tableau 1.

The 0-1 vector \bar{x} whose support is $\{2,3,5,12,13,17\}$ is a cover, satisfying with equality all the inequalities except for 1 and 8, which are oversatisfied. To apply the Bellmore-Ratliff procedure, one associates with \bar{x} an involutory basis. The variables x_2, x_3, x_5 can be basic only in rows 3, 4 and 6 respectively. Since rows 1 and 8 are slack, the variables

x_{12} and x_{13} can be basic only in rows 11 and 10 respectively. Finally, x_{17} can be basic in any of the 4 rows 2, 5, 7, 9; and accordingly there are 4 involutory bases that can be associated with \bar{x} . We will denote them by B_2, B_5, B_7 and B_9 , according as x_{17} is basic in row 2, 5, 7 or 9 respectively. The basis B_2 (after row permutations) is shown in Tableau 2.

	2	3	5	12	13	17	25	27	29	21	28
3	1										
4		1									
6			1								
11				1							
10					1						
2						1					
5						1	-1				
7						1		-1			
9						1			-1		
1					1	1				-1	
8				1	1						-1

Tableau 2.

The variables x_{21}, \dots, x_{31} are slacks. The basis B_5 can be obtained from B_2 by interchanging rows 5 and 2, and replacing the slack variable x_{25} in row 5, by x_{22} in row 2. The other two bases can be obtained in an analogous way. The 4 cutting planes that can be obtained by the Bellmore-Ratliff procedure, depending on which basis is used, are

$$\begin{aligned}
 x_1 + x_6 + x_9 + x_{10} + x_{15} + x_{16} + x_{18} + x_{20} &\geq 1, && \text{from } B_2 \\
 x_4 + x_6 + x_9 + x_{10} + x_{11} + x_{19} &\geq 1, && \text{from } B_5 \\
 x_6 + x_7 + x_{10} + x_{15} + x_{19} + x_{20} &\geq 1, && \text{from } B_7 \\
 x_6 + x_8 + x_{10} + x_{14} + x_{18} + x_{20} &\geq 1, && \text{from } B_9.
 \end{aligned}$$

On the other hand, using the conditional bound approach, we construct (by inspection or a heuristic) the dual vector

$$u = (0,1,1,1,1,1,2,0,1,2,2)$$

which, together with the associated reduced cost vector

$$s = (2,0,0,2,0,0,0,1,1,0,1,1,1,1,0,0,2,0,1)$$

satisfies the condition (1).

The cover x whose support is $\{2,3,5,12,13,17\}$ yields $z_U = c\bar{x} = 14$; and the dual vector u yields the lower bound $ue = 12$.

Since $z_U - ue = 2$, $Q_0 = \{j \in N \mid s_j \geq 2\} = \{1,4,18\}$, and thus (Remark 2.1) every cover better than \bar{x} satisfies $x_1 = x_4 = x_{18} = 0$. Hence we replace N by $N \setminus \{1,4,18\}$. Further, to apply Theorems 2.1 and 3.1, we set $p = 2$, with $v_1 = s_{12}$ and $v_2 = s_{13}$,

$$\begin{aligned} s_{12} + s_{13} &= 2 \\ &\geq z_U - ue. \end{aligned}$$

Then, using (for instance) the conditional inequalities (and the matrix C) defined by

$$Q_1 = \{8,12,14,20\}, \quad Q_2 = \{9,11,13,15\}$$

we obtain the disjunction

$$x_8 = x_{12} = x_{14} = x_{20} = 0 \quad \vee \quad x_9 = x_{11} = x_{13} = x_{15} = 0.$$

Applying Theorem 3.1 with $h(1) = 11$, $h(2) = 10$, we have (using the newly defined set N)

$$N_{h(1)} \setminus Q_1 = \{6,19\}$$

$$N_{h(2)} \setminus Q_2 = \{10,16,19\}$$

and thus

$$W = \{6, 10, 16, 19\} .$$

We have obtained the cut

$$x_6 + x_{10} + x_{16} + x_{19} \geq 1$$

which has only 4 positive coefficients, whereas each of the involutory basis cuts has at least 6.

The above inequality cuts off \bar{x} . This is due to the way we chose the components of v and the row indices $h(i) \in M$, as will be shown in the next section. If we drop the requirement for a specified cover to be cut off, we can obtain inequalities which are "stronger" in the sense of having fewer positive coefficients. By a judicious choice of the column indices $j(i)$ for which we set $v_i = s_{j(i)}$, and the row indices $h(i)$, one can generate the cuts with the fewest possible positive coefficients. Thus, for instance, the choice $v_1 = s_{13}$, $v_2 = s_9$, and $h(1) = 8$, $h(2) = 5$, yields the valid inequality

$$x_{17} + x_{19} \geq 1 ;$$

whereas $v_1 = s_{13}$, $v_2 = s_{14}$ and $h(1) = 8$, $h(2) = 4$ yields

$$x_3 + x_{19} \geq 1 .$$

4. Some Properties of Cuts from Conditional Bounds

In order to obtain a conditional bound, and thus to be able to generate the type of constraints discussed in the previous section, one needs a feasible solution u to (D), whose associated reduced costs satisfy (8). This requirement is easy to meet. The next theorem and its Corollary give a broad sufficient condition for u to satisfy them.

Let $S(\bar{x})$ denote the support of \bar{x} .

Theorem 4.1. Let \bar{x} be a cover for (SC) and let \bar{u} and \bar{s} satisfy (1) and (7). If \bar{u} also satisfies

$$\bar{u}(A\bar{x} - 1) = 0,$$

then

$$(12) \quad \sum_{j \in S(\bar{x})} \bar{s}_j = c\bar{x} - \bar{u}e.$$

Proof. Let

$$S^+ = \{j \in S(\bar{x}) \mid \bar{s}_j > 0\}$$

and consider the pair of dual linear programs

$$(L_1) \quad \min \{cx \mid Ax \geq e, x_j \geq 1, j \in S^+, x_j \geq 0, j \in N \setminus S^+\}$$

and

$$(D_1) \quad \max \{ue + \sum_{j \in S^+} s_j \mid ua_j + s_j = c_j, j \in N; u \geq 0, s \geq 0\}.$$

Clearly, \bar{x} is a feasible solution to (L_1) , and (\bar{u}, \bar{s}) is a feasible solution to (D_1) . Further, \bar{x} and (\bar{u}, \bar{s}) satisfy the complementary slackness conditions

$$\begin{aligned} (a^i_1 \bar{x} - 1)u_i &= 0, \quad i \in M \\ (x_j - 1)s_j &= 0, \quad j \in S^+ \\ x_j s_j &= 0, \quad j \in N \setminus S^+ \end{aligned}$$

Hence \bar{x} and (\bar{u}, \bar{s}) are optimal solutions to (L_1) and (D_1) , respectively, which implies the equality of the two objective function values.

Q.E.D.

An immediate consequence of Theorem 4.1 is the following.

Let

$$T(\bar{x}) = \{i \in M \mid a^i \bar{x} = 1\}.$$

Corollary 4.1.1. Let \bar{x} be a cover, and let \bar{u} and \bar{s} satisfy (1) and (7). If \bar{u} also satisfies

$$(13) \quad \bar{u}_i = 0, \quad \forall i \in M \setminus T(\bar{x}),$$

then there exists $\{j(1), \dots, j(p)\} \subseteq S^+$ such that v defined by $v_i = \bar{s}_{j(1)}$, $i = 1, \dots, p$, together with \bar{u} , satisfies (5).

Proof. If \bar{u} satisfies (13), then together with \bar{x} it satisfies the complementarity condition of Theorem 4.1, and therefore (12) holds.

From (12), v_i as defined in the Corollary, together with \bar{u} , satisfies (5).

Q.E.D.

Thus, any feasible solution to (D) that has positive components only for $i \in T(\bar{x})$ produces a vector v which satisfies (5). A procedure of the type outlined in section 2 can be used to find a matrix C which satisfies the other requirements of Theorem 2.1. Choosing appropriate rows $h(i) \in M$ will then produce a cut of the family (10) defined in Theorem 3.1.

From the point of view of having a finitely convergent procedure, it is essential that the inequalities generated be not only valid but also new, i.e., no cut should be generated twice.

Given two inequalities of the form (10) with associated index sets W_1 and W_2 , the first inequality implies the second one if and only if $W_1 \subseteq W_2$. We will say that an inequality (10) is new, if it is not implied by any inequality of the current problem (SC).

Theorem 4.2. The inequality

$$(10) \quad \sum_{j \in W} x_j \geq 1$$

of Theorem 3.1 is new if and only if $N \setminus W$ is the support of a cover.

Proof. If $N \setminus W$ is the support of a cover, say \bar{x} , then \bar{x} violates (10), whereas it satisfies all the inequalities of the current problem (SC). Hence (10) is new. Conversely, if $N \setminus W$ does not define a cover, then there exists a row $i \in M$ such that $N_i \cap (N \setminus W) = \emptyset$, i.e., $N_i \subseteq W$. But then the cut (10) defined by W is either identical to, or implied by, the i -th inequality of (SC). Q.E.D.

Theorem 4.2 gives a necessary and sufficient condition for an inequality (10) to cut off at least one cover. Next we give a sufficient condition for an inequality (10) to cut off a specified prime cover \bar{x} .

As before we will denote

$$M_j = \{i \in M \mid a_{ij} = 1\}, \quad j \in N.$$

Remark 4.1. If \bar{x} is a prime cover for (SC), then

$$M_j \cap T(\bar{x}) \neq \emptyset, \quad \forall j \in S(\bar{x}).$$

Proof. Follows from the definition of $T(\bar{x})$ and that of a cover prime.

Theorem 4.3. Let \bar{x} be a prime cover for (SC), let z_U be a known upper bound on the value of (SC), and let u and s satisfy (1) and (7). Further, let

$$(9) \quad \sum_{j \in S} s_j \geq z_U - u$$

for some $S \subseteq S(\bar{x})$, $S = \{j(1), \dots, j(p)\}$. Define

$$(14) \quad v_i = s_{j(i)}, \quad i = 1, \dots, p,$$

and let C be any $p \times n$ 0-1 matrix satisfying conditions (3), (4) and

$$(15) \quad c_{hj(i)} = \begin{cases} 1 & h = i \\ 0 & h \neq i \end{cases} \quad i = 1, \dots, p.$$

Finally, let $Q_i = \{j \in N \mid c_{ij} = 1\}$ and

$$W = \bigcup_{i=1}^p [N_{h(i)} \setminus Q_i],$$

where

$$(16) \quad h(i) \in T(\bar{x}) \cap M_{j(i)}, \quad i = 1, \dots, p.$$

Then the inequality

$$(10) \quad \sum_{j \in W} x_j \geq 1$$

is satisfied by every cover x such that $cx < z_U$, and cuts off \bar{x} .

Proof. By assumption, the pair u, s satisfies (1) and (7), while the matrix C satisfies (3) and, together with the vector v defined by (14), satisfies (4). From (9), u and v satisfy (5). The existence of such a pair C, v follows from (9) and Theorem 2.2. The additional requirement (15) on C can always be met. Hence the pair C, v defined in the Theorem exists and satisfies the requirements of Theorems 2.1 and 3.1.

Further, since $j(i) \in S(\bar{x})$, we have $T(\bar{x}) \cap M_{j(i)}, i = 1, \dots, p$ (Remark 4.1), i.e., there exists a set of row indices $h(i), i = 1, \dots, p$, satisfying condition (16). From that condition, and the definition of $T(\bar{x})$, we have

$$S(\bar{x}) \cap N_{h(i)} = \{j(i)\}, \quad i = 1, \dots, p.$$

On the other hand, from (15), $j(i) \in Q_i$, $i = 1, \dots, p$. Hence

$$S(\bar{x}) \cap [N_{h(i)} \setminus Q_i] = \emptyset, \quad i = 1, \dots, p,$$

i.e. $S(\bar{x}) \cap W = \emptyset$ and the inequality (10) cuts off \bar{x} .

Q.E.D.

The results of the last two sections can be used to generate a family of inequalities whose members are different from each other and from the initial constraints of (SC), and are satisfied by every cover whose value is smaller than z_U . Since the family of such cuts is of considerable size, one would like to be able to generate some of the "stronger" members, according to some reasonable measure of strength. This will be discussed in the next section. First, however, we examine another question related to the properties discussed above.

If the set Q_0 defined in Remark 2.1 is nonempty, then the variables indexed by Q_0 can be set to 0. Since the vector u (and the associated s) used in the definition of Q_0 , is not required to have any other property but to satisfy (1), the question arises as to which vector u is more likely to produce a nonempty set Q_0 , or more generally, a larger set Q_0 . While this question is hard to answer in general, there is a clear answer for the case when the choice is between two vectors u^1, u^2 such that $u^1 \leq u^2$.

Theorem 4.4. For $k = 1, 2$, let u^k satisfy (1), and define

$$s_j^k = c_j - \sum_{i \in M} u_i^k a_{ij}, \quad j \in N,$$

and

$$Q_0^k = \{j \in N \mid s_j^k \geq z_U - u^k e\}.$$

Then $u^1 \leq u^2$ implies $Q_0^1 \subseteq Q_0^2$.

Proof. If $j \in Q_0^1$, then

$$s_j^1 = c_j - \sum_{i \in M} u_i^1 a_{ij}$$

$$\geq z_U - u^1 e,$$

i.e.,

$$c_j + \sum_{i \in M \setminus M_j} u_i^1 \geq z_U;$$

and if $u^1 \leq u^2$, the last inequality implies

$$c_j + \sum_{i \in M \setminus M_j} u_i^2 \geq z_U,$$

i.e.,

$$s_j^2 = c_j - \sum_{i \in M} u_i^2 a_{ij}$$

$$\geq z_U - u^2 e,$$

hence $j \in Q_0^2$.

Q.E.D.

Thus, the set Q_0 defined with respect to a vector u satisfying (1) is always contained in the set Q_0 defined with respect to any vector \hat{u} obtained from u by increasing some of its components (while maintaining $\hat{u}A \leq c$). Therefore, in generating the set Q_0 , one should always use a "maximal" dual vector u , i.e., one whose components cannot be increased without decreasing some component or violating $uA \leq c$.

5. Generating Cuts

In this section we discuss two procedures for generating conveniently strong members of the family of cuts introduced above. Since all inequalities have a right hand side of 1 and left hand side coefficients

of 0 or 1, we will use as a measure of strength the number of positive coefficients (the smaller the number, the stronger the inequality). The fact that an inequality A is stronger than B in this sense, i.e., has fewer positive coefficients, does not imply of course that A dominates B or implies B.

Procedure CUT generates a member of the subfamily characterized in Theorem 4.3, i.e., an inequality which is satisfied by every solution x such that $cx < z_U$, and which also cuts off a specified prime cover \bar{x} . The strength of an inequality (10), i.e., the size of the set W , depends on the integer p and the size of the sets $N_{h(i)} \setminus Q_1$, $i = 1, \dots, p$, of Theorem 3.1. To have p conveniently small, the procedure sets $v_1 = s_j$ for a sequence of indices $j(i)$, $i = 1, \dots, p$, corresponding to the largest reduced costs s_j , $j \in S(\bar{x})$. Each row index $h(i)$, $i = 1, \dots, p$, is of course chosen from $T(\bar{x}) \cap M_{j(i)}$, as prescribed by Theorem 4.3. Further, in order to have W as small as possible, the sequence of row indices $h(i)$ is chosen so as to come as close as possible at each step $k \in \{1, \dots, p\}$ to minimizing the set $W_k \setminus W_{k-1}$, where $W_0 = \emptyset$ and

$$W_k = \bigcup_{i=1}^k [N_{h(i)} \setminus Q_1] \quad , \quad k = 1, \dots, p.$$

Since $|S| = 1$ implies (Remark 2.1) that some variable can be permanently set to zero, we assume that this has already been done for all variables indexed by Q_0 , and thus there are no singleton sets S satisfying (9).

Let M and N be the row and column index sets of the current problem (SC), let \bar{x} be a prime cover for (SC), and let $S(\bar{x}) = \{j \in N \mid \bar{x}_j = 1\}$, $T(\bar{x}) = \{i \in M \mid \bar{x}_i = 1\}$. Further, let u and s satisfy (1), (7) and (9) for $S = S^+$, where

$$S^+ = \{j \in S(\bar{x}) \mid s_j > 0\}.$$

CUT

Step 0. Initialize $W_0 = \emptyset$, $K_1 = S^+$, $y_1 = ue$, $s^1 = s$. Let $t \leftarrow 1$ (iteration counter) and go to 1:

Step 1. Define

$$v_t = \min\{z_U - y_t, \max_{j \in K_t} s_j^t\}, \quad K_t^* = \{j \in K_t \mid s_j^t = v_t\}$$

$$P_t = \{j \in N \mid s_j^t \geq v_t\}, \quad M(t) = \bigcup_{j \in K_t^*} M_j.$$

Choose $i(t)$ such that

$$|N_{i(t)} \setminus (P_t \cup W_{t-1})| = \min_{i \in T(\bar{x}) \cap M(t)} |N_i \setminus (P_t \cup W_{t-1})|$$

(breaking ties arbitrarily), and let

$$\{j(t)\} = N_{i(t)} \cap K_t^*.$$

Then define

$$W_t = W_{t-1} \cup [N_{i(t)} \setminus P_t], \quad y_{t+1} = y_t + v_t.$$

If $y_{t+1} \geq z_U$, go to 2. Otherwise, let

$$s_j^{t+1} = \begin{cases} s_j^t - v_t & j \in N_{i(t)} \cap P_t \\ s_j^t & \text{otherwise,} \end{cases}$$

$K_{t+1} = K_t \setminus \{j(t)\}$, set $t \leftarrow t + 1$, and go to 1.

Step 2. Define $W = W_t$ and add to the constraints of (SC) the inequality

$$\sum_{j \in W} x_j \geq 1.$$

Theorem 5.1. In $|S^+|$ iterations, procedure CUT generates an inequality satisfied by every cover x such that $cx < z_U$, and violated by \bar{x} .

Proof. Since $K_1 = S^+$ decreases by one element at each iteration, the procedure ends in $|S^+|$ iterations. Next we show that the inequality generated by the procedure satisfies the conditions of Theorem 4.3.

The vectors u , s satisfy (1) and (7). The vector v used in the procedure satisfies condition (14) of Theorem 4.3, while the matrix C used in the procedure (not stated explicitly) is defined by

$$c_{ij} = \begin{cases} 1 & j \in Q_t = P_t \cap N_{i(t)} \\ 0 & \text{otherwise} \end{cases}$$

This satisfies requirement (3) and also (15), since $\{j(t)\} = N_{i(t)} \cap K_t^*$ is unique [from $i(t) \in T(\bar{x})$] and $j(t) \neq j(t-1)$, $t = 2, \dots, p$. From the definition of the sets P_t , it follows that C and v satisfy (4). The stopping rule $y_{t+1} \geq z_U$ guarantees that the set $S = \{j(1), \dots, j(p)\}$, where p is the number of iterations, satisfies (9). Finally, the choice of $i(t)$ in step 1 insures that condition (16) is met. Q.E.D.

Note that the above inequality was obtained for a given value of z_U . If later the upper bound z_U is improved, i.e., replaced by $z_U' < z_U$, the cut can be strengthened. Namely, we have

Corollary 5.1.1. Suppose

$$(17) \quad \sum_{j \in W_p} x_j \geq 1$$

is a cut obtained in p iterations of CUT for an upper bound z_U , and subsequently z_U is replaced by $z'_U < z_U$. Then every cover x for which $cx < z'_U$ satisfies

$$(18) \quad \sum_{j \in W_{p-k}} x_j \geq 1,$$

where k is the greatest integer, if it exists, such that

$$(19) \quad \sum_{i=1}^k v_{p+1-i} \leq z_U - z'_U,$$

or else $k = 0$.

Proof. At the end of p iterations of CUT, we have

$$(20) \quad y_{p+1} = ue + \sum_{i=1}^p v_i \geq z_U,$$

which implies that the inequality (17) is satisfied by every cover better than the one associated with z_U . Subtracting (19) from (20) yields

$$y_{p-k+1} = ue + \sum_{i=1}^{p-k} v_i \geq z'_U,$$

which implies that the inequality (18) is satisfied by every cover better than the one associated with z'_U . Q.E.D.

The information required in order to be able to strengthen a cut generated by CUT 1 whenever z_U is improved, is the sequence of pairs $(W_1, v_1), (W_2 \setminus W_1, v_2), \dots, (W_p \setminus W_{p-1}, v_p)$.

6. Generating Prime Covers

Any algorithm based on the cutting planes introduced in this paper needs to generate repeatedly prime covers for the current problem (SC), and feasible solutions to the dual (D) of the linear program associated with (SC). In this section we discuss some heuristics for generating reasonably good prime covers for (SC) and feasible solutions to (D), "good" in the sense of giving tight bounds.

The procedure PRIMAL 1, to be described below, is a kind of "greedy" algorithm, which selects at each step the locally most promising column, and in one pass produces a (usually remarkably good) prime cover \bar{x} .

As before, let M and N be the row and column index set of the current problem (SC), with $M_j = \{i \in M | a_{ij} = 1\}$ and $N_i = \{j \in N | a_{ij} = 1\}$. Further, let z_U be an upper bound and z_L a lower bound on the value of (SC).

PRIMAL 1

Step 0. Initialize $R_1 = m$, $S_1 = \emptyset$, $t \leftarrow 1$, and go to 1.

Step 1. If $R_t = \emptyset$, set $S = S_t$ and go to 2. Otherwise, define

$$I(t) = \{i \in R_t \mid |N_i| = \min_{h \in R_t} |N_h|\}, \quad N_{I(t)} = \bigcup_{i \in I(t)} N_i.$$

If $N_{I(t)} = \emptyset$, stop; the cover corresponding to z_U is optimal.

Otherwise, choose $j(t) \in N_{I(t)}$ such that

$$c_{j(t)} / |R_t \cap M_{j(t)}| = \min_{j \in N_{I(t)}} \{c_j / |R_t \cap M_j|\}$$

if this minimum is unique. If it is not, and it is attained for $j \in J(t)$, $|J(t)| > 1$, choose $j(t)$ such that

$$|R_t \cap M_{j(t)}| = \max_{j \in J(t)} |R_t \cap M_j|$$

and if there are further ties, break them by maximizing $|M_j|$.

Then set $S_{t+1} = S_t \cup \{j(t)\}$, $R_{t+1} = R_t \setminus M_{j(t)}$, $t \leftarrow t + 1$, and go to 1.

Step 2. Consider the elements $i \in S$ in order of decreasing costs c_i , and if

$$\bigcup_{j \in S \setminus \{i\}} M_j = M,$$

remove i from S . If all $i \in S$ have been considered, \bar{x} defined by $\bar{x}_j = 1$, $j \in S$, $\bar{x}_j = 0$ otherwise, is a prime cover. If $c\bar{x} < z_U$, set $z_U = c\bar{x}$ and stop. If $z_U \leq z_L$, \bar{x} is optimal.

The logic behind the choice of $j(t)$ in Step 1 is that, the smaller the number of columns that can be used to cover a particular row, say i , the higher the price to be paid for not covering optimally row i is likely to be. Therefore the procedure first chooses a row $i(t)$ with a minimum number of 1's in it, then among the columns which can be used to cover row $i(t)$, it selects one with minimum cost per row covered. Ties are broken by choosing the column which covers the largest number of new rows (at the same unit cost), or, if there still is a tie, the column which covers the largest total number of rows. When all the rows have been covered ($R_t = \emptyset$), step 2 checks whether the cover is prime, and makes it prime if necessary by dropping redundant columns in order of decreasing cost.

The above procedure generates a prime cover for the current problem (SC) by starting from scratch. Though this procedure is relatively cheap, it is nevertheless often desirable to have a procedure which starts with a partial cover rather than from scratch. For instance, if $S(\bar{x}) \cap Q_0 \neq \emptyset$, where $S(\bar{x})$ is the support of the current prime cover \bar{x} and Q_0 is the set of Remark 2.1, i.e., if some variables which are at 1 in the current cover can be fixed at 0, it is desirable to find a new cover starting with $S = S(\bar{x}) \setminus Q_0$. Also, when \bar{x} is a prime cover for the current problem (SC) but ceases to be a cover because of the addition to (SC) of new cuts, \bar{x} can be used as a partial cover to start the procedure for finding a new cover. PRIMAL 2 is a modified version of PRIMAL 1, which differs from the latter only in the initialization step, where $S_1 = \emptyset$ is replaced by $S_1 = S(\bar{x})$, the support of the partial cover that we wish to start with.

Finally, it often happens that a good dual solution u is found, i.e., one which yields a high lower bound ue , but which, together with the associated reduced cost vector s , does not satisfy (9) for $S = S(\bar{x})$, where \bar{x} is the current cover. If (9) cannot be satisfied for any set $S \subseteq N$, then of course we have no choice other than to modify u and s , i.e., decrease some components of u , which will then increase (usually by more) the sum of the reduced costs. If, however, u and s satisfy (9) for $S = N$, then it is worth trying to change the cover rather than the vector u ; for if a cover (any cover) \hat{x} can be found such that (9) holds for $S = S(\hat{x})$, then a cut can be generated whose strength depends only on u , s and z_u , but not on $c\hat{x}$. In other words, no

matter how bad the new cover \hat{x} is in terms of the associated objective function, if it makes (9) hold for u, s then it provides for a better starting point for a cut than could be obtained by keeping the old cover \bar{x} and changing u and s .

The procedure PRIMAL 3 is meant for situations like this. It starts with the old cover \bar{x} , and successively introduces variables x_j such that $s_j > 0$. Every time such a variable is introduced, another variable x_h , such that $s_h = 0$, is dropped. Furthermore, x_j is chosen so as to cover at least one row not covered by other variables with positive reduced cost, and x_h is chosen so as to leave at least one row covered only by x_j . These choice rules are intended to keep the cover prime if possible. When this procedure cannot be continued further, then the partial cover at hand is completed by using as few columns as possible, again in an attempt to keep the cover prime. The cover obtained this way is then made prime, if necessary, by removing some variable(s) x_j , preferably such that $s_j = 0$.

As before, let $T(\bar{x}) = \{i \in M \mid a^i \bar{x} = 1\}$, $S^+ = \{j \in S(\bar{x}) \mid s_j > 0\}$, and denote $N^+ = \{j \in N \mid s_j > 0\}$. We assume that

$$\sum_{j \in N^+} s_j \geq z_U - ue.$$

PRIMAL 3

Step 0. Initialize $S_1 = S(\bar{x})$, $K_1 = N^+ \setminus S^+$, $R_1 = M \setminus \bigcup_{j \in S^+} M_j$, and go to 1.

Step 1. If $R_t = \emptyset$, set $S = S_t$ and go to 3. If $R_t \neq \emptyset$ but $M_j \cap R_t = \emptyset$, $\forall j \in K_t$, or $s_j > 0$, $\forall j \in S_t$, set $R_t = M \setminus \bigcup_{j \in S_t} M_j$ and go to 2.

Otherwise choose $j(t,1) \in K_t$ and $j(t,2) \in S_t \setminus N^+$ such that

$$|M_{j(t,1)} \cap R_t| = \max_{j \in K_t} |M_j \cap R_t|$$

and

$$|M_{j(t,2)} \cap M_{j(t,1)} \cap R_t| = \max_{j \in S_t \setminus N^+} |M_j \cap M_{j(t,1)} \cap R_t|.$$

Then $S_{t+1} = S_t \cup \{j(t,1)\} \setminus \{j(t,2)\}$, $K_{t+1} = K_t \setminus \{j(t,1)\}$, $R_{t+1} = R_t \setminus M_{j(t,1)}$, $t \leftarrow t + 1$, and go to 1.

Step 2. If $R_t = \emptyset$, set $S = S_t$ and go to 3. If $R_t \neq \emptyset$ but $M_j \cap R_t = \emptyset$, $\forall j \in N \setminus S_t$, stop; the cover corresponding to z_U is optimal. Otherwise, choose $j(t)$ such that

$$|M_{j(t)} \cap R_t| = \max_{j \in N \setminus S_t} |M_j \cap R_t|,$$

set $S_{t+1} = S_t \cup \{j(t)\}$, $R_{t+1} = R_t \setminus M_{j(t)}$, $t \leftarrow t + 1$, and go to 2.

Step 3. Consider first the elements $i \in S$ such that $s_i^t = 0$, then the remaining $i \in S$, and if

$$\bigcup_{j \in S \setminus \{i\}} M_j = M,$$

remove i from S . If all $i \in S$ have been considered, \bar{x} defined by $\bar{x}_j = 1$, $j \in S$, $\bar{x}_j = 0$ otherwise, is a prime cover. If

$$\sum_{j \in S} s_j \geq z_U - ue,$$

then s can be used to generate a cut. Otherwise u and s have to be modified.

Next we turn to the problem of generating "good" feasible solutions u to (D).

7. Generating Feasible Solutions to (D)

As already mentioned, the quality of a solution u to (D) is given by the quality of the lower bound ue on the value of (SC), provided that u and the associated reduced costs s_j satisfy (9) for $S = S(\bar{x})$. Among two feasible solutions to (D) with the same value ue , the one with a higher value of $\sigma = \sum_{j \in N} s_j$ is likely to produce a disjunction (6) with fewer terms and more elements in each term, i.e., a stronger cut. Hence the purpose of our heuristics will be to approximate as much as possible the lexicographic maximum of the two-component vector (ue, σ) .

This is what the procedure DUAL 1, to be described below, tries to accomplish. It successively assigns values to components of u , which are maximal subject to the dual constraints and the earlier value-assignments. The order in which the values are assigned is crucial to the quality of the resulting solution, and DUAL 1 considers the rows of A in order of increasing N_i . Once a value has been assigned to each component of u , the set Q_0 of Remark 2.1 is identified and the corresponding variables are set to zero. If u and the associated s satisfy (9) for $S = S(\bar{x})$, then s can be used to generate a cut. Otherwise either the cover must be changed as discussed in the previous section (see PRIMAL 3), or the vector u must be adjusted (see DUAL 3 below).

Let $S(\bar{x})$ be the support of a prime cover for the current problem (SC), and let z_U and z_L be an upper and a lower bound, respectively, on the value of (SC).

DUAL 1

Step 0. Initialize $R_1 = M$, $u^1 = 0$, $s^1 = c$, $t \leftarrow 1$, and go to 1.

Step 1. If $R_t = \emptyset$, go to 3. Otherwise choose $i(t)$ such that

$$N_{i(t)} = \min_{i \in R_t} |N_i|,$$

and set

$$u_i^{t+1} = \begin{cases} \min_{j \in N_{i(t)}} s_j^t & i = i(t) \\ u_i^t & \text{otherwise,} \end{cases} \quad s_j^{t+1} = \begin{cases} s_j^t + u_i^t - u_i^{t+1} & j \in N_{i(t)} \\ s_j^t & \text{otherwise.} \end{cases}$$

Define

$$F_t = \{j \in N_{i(t)} \mid s_j^{t+1} = 0\}, \quad M(t) = \bigcup_{j \in F_t} M_j$$

set $R_{t+1} = R_t \setminus M(t)$, $t \leftarrow t + 1$, and go to 1.

Step 2. If $u^t e > z_L$, set $z_L = u^t e$, store s^t in place of the vector s associated with the previous bound z_L , and stop. If $z_L \leq z_U$, the cover associated with z_U is optimal.

As already mentioned, the choice of $i(t)$ is crucial for the efficiency of the procedure. Since the rule of Step 1 usually leads to ties, one may want to think of secondary criteria. Several such criteria are currently being tested. One which has been found to frequently improve the quality

of the solutions is to give preference to $i \in T(\bar{x})$ over $i \in M \setminus T(\bar{x})$ in choosing $i(t)$ in Step 1.

While DUAL 1 is a computationally cheap one pass procedure, it starts from scratch every time a new pair of vectors u, s is to be found. Next we state a procedure (DUAL 2) which starts from a dual vector u and modifies it after the addition of a new cut and the finding of a new cover, by setting to 1 the variable associated with the new inequality, and adjusting the values of some other variables.

Let m be the index of the last inequality added to (SC) after generating the current dual vector $u = (u_1, \dots, u_{m-1})$ and the associated reduced cost vector s . Further, let \bar{x} be a cover for the current (SC), $S(\bar{x})$ its support, and $T(\bar{x}) = \{i \in M \mid a_i^1 \bar{x} = 1\}$.

DUAL 2

Step 0. Initialize

$$u_i^1 = \begin{cases} 1 & i = m \\ 0 & i \in [M \setminus T(\bar{x})] \\ u_i & \text{otherwise,} \end{cases}$$

$s^1 = c - u^1 A$, and $K_1^- = \{j \in N \mid s_j^1 < 0\}$. Set $t \leftarrow 1$ and go to 1a.

Step 1a. If $K_t^- = \emptyset$, set $K_t^0 = \{j \in N \mid s_j^t = 0\}$, $R_t = \{i \in M \mid N_i \cap K_t^0 = \emptyset\}$, and go to 1. Otherwise, choose $i(t)$ such that

$$|N_{i(t)} \cap K_t^-| = \max_{i \in M \setminus \{m\} \mid u_i > 0} |N_i \cap K_t^-|,$$

set

$$u_i^{t+1} = \begin{cases} u_i^t - 1 & i = i(t) \\ u_i^t & \text{otherwise} \end{cases} \quad s_j^{t+1} = \begin{cases} s_j^t + 1 & j \in N_{i(t)} \\ s_j^t & \text{otherwise.} \end{cases}$$

Then set $K_{t+1}^- = \{j \in N | s_j^{t+1} < 0\}$, $t \leftarrow t + 1$, and go to 1a.

Steps 1-2 are the same as in DUAL 1.

Finally, we have to deal with the situation when a pair u, s has to be adjusted so as to satisfy (9) for some $S = S(\bar{x})$, where \bar{x} is a given prime cover. The following procedure is meant to achieve this.

DUAL 3

Step 0. Initialize $u^1 = u$, $s^1 = s$, $U_t = \{i \in M \setminus T(\bar{x}) | u_i^1 > 0\}$. Set $t \leftarrow 1$ and go to 1.

Step 1. Choose $i(t)$ such that

$$|N_{i(t)} \cap S(\bar{x})| = \max_{i \in U_t} |N_i \cap S(\bar{x})|.$$

Then set

$$u_i^{t+1} = \begin{cases} u_i^t - 1 & i = i(t) \\ u_i^t & \text{otherwise,} \end{cases} \quad s_j^{t+1} = \begin{cases} u_j^t + 1 & j \in N_{i(t)} \\ s_j^t & \text{otherwise.} \end{cases}$$

If u^{t+1} and s^{t+1} satisfy (9) for $S = S(\bar{x})$, stop: $s = s^{t+1}$ can be used in CUT 1. Otherwise, set $U_{t+1} = U_t \setminus \{i(t)\}$, $t \leftarrow t + 1$, and go to 1.

The heuristics discussed here find reasonably good dual vectors u at a low computational cost. Another possibility is, of course, to solve (D) by the simplex method. While this is more costly, only computational testing can show whether the improvement in the bound ue is worth the extra computational effort.

8. Algorithms

In this section we present two algorithms for solving set covering problems by cutting planes from conditional bounds.

They use as ingredients the procedures discussed in the last three sections (for generating prime covers, feasible dual vectors and valid cutting planes), and the following three subroutines for updating cuts and fixing variables whenever possible.

STRENGTHEN (to be used whenever PRIMAL replaces z_U^{old} by z_U^{new}).

In every cut $\sum_{j \in W_p} x_j \geq 1$, replace W_p by W_{p-k} , where k is the greatest integer such that $v_p + \dots + v_{p-k+1} \leq z_U^{\text{old}} - z_U^{\text{new}}$.

TEST 1 (to be used after STRENGTHEN).

Let s be the reduced cost vector associated with z_L , and let $S(\bar{x})$ be the support of the last cover \bar{x} . Define

$$Q_0 = \{j \in N \mid s_j \geq z_U^{\text{new}} - z_L\},$$

set $x_j = 0$, $j \in Q_0$, and $N \leftarrow N \setminus Q_0$. If $N \subseteq S(\bar{x})$, the cover associated with z_U^{new} is optimal.

TEST 2 (to be used after DUAL).

Same as TEST 1, except for the fact that s , z_U^{new} and z_L are replaced by s^t , z_U and u^t respectively, where u^t is the last dual vector obtained, s^t is the reduced cost vector associated with u^t , and z_U is the current upper bound.

Algorithm 1 starts by initializing the upper bound z_U at $\sum c_j$ and the lower bound z_L at 0. A typical iteration consists of the following sequence of steps.

- A. Use PRIMAL 1 or 2 to generate a prime cover \bar{x} . If z_U is improved, go to C1; otherwise go to B1.
- B1. Use DUAL 1 to generate a dual vector u and go to C2.
- B2. Use DUAL 3 to adjust u and s and go to D.
- C1. Apply TEST 1 to fix variables. If $N \subseteq S(\bar{x})$, stop. Otherwise, if \bar{x} is still a cover, go to B1; else go to A.
- C2. Apply TEST 2 to fix variables. If $N \subseteq S(\bar{x})$, stop. Otherwise, if \bar{x} is still a cover and (9) holds for $S = S(\bar{x})$, go to D; if (9) does not hold for $S = S(\bar{x})$, go to B2; and if \bar{x} is not any more a cover, go to A.
- D. Use CUT to generate a valid inequality which cuts off the last cover. Add the new inequality to (SC) and go to A.

Algorithm 1 seems the simplest possible way to use the cuts from conditional bounds. Algorithm 2, stated below, is a more sophisticated procedure based on the same approach, which (a) strengthens the cuts whenever z_U is improved; (b) avoids starting from scratch every time a new cover or a new dual solution is generated by using the "updating" procedures PRIMAL 2 and DUAL 2, with a periodic return to PRIMAL 1

and DUAL 1; and (c) uses PRIMAL 3 whenever z_L is improved and (9) does not hold for the current cover, to find a prime cover \bar{x} such that u and s satisfy (9) for $S = S(\bar{x})$. Since the chances for finding a better cover than the current best one are steadily diminishing during the procedure, the parameter α defining the number of iterations after which we return to PRIMAL 1 is doubled every time the latter procedure is applied; while the parameter β defining the frequency of return to DUAL 1 is kept constant.

After initializing z_U , z_L , α and β , a typical iteration of Algorithm 2 consists of some of the following steps.

A1. Use PRIMAL 1 to generate a prime cover \bar{x} , and double the value of α . If z_U is improved, go to C1; otherwise to B1.

A2. Use PRIMAL 2 to generate a prime cover \bar{x} . If z_U is improved, go to C1; otherwise go to B2, or (if no cut was added after obtaining last vector u , or B2 was used β times since last use of B1) to B1.

A3. Use PRIMAL 3 to generate a prime cover \bar{x} . If the last u and s satisfying (9) for $S = S(\bar{x})$, go to D; otherwise go to B3.

B1. Use DUAL 1 to generate a dual vector u and go to C2.

B2. Use DUAL 2 to generate a dual vector u and go to C2.

B3. Use DUAL 3 to adjust u and s and go to D.

C1. Use STRENGTHEN to strengthen the cuts, and TEST 1 to fix variables. If $N \subseteq S(\bar{x})$, stop. Otherwise, if \bar{x} is still a cover, go to B1; else go to A2.

C2. Use TEST 2 to fix variables. If $N \subseteq S(\bar{x})$, stop. Otherwise, if \bar{x} is still a cover and (9) holds for $S = S(\bar{x})$, go to D; if (9) does not hold for $S = S(\bar{x})$ but holds for $S = N$, go to A3; and if (9) does not hold for $S = N$, go to B1; finally, if \bar{x} is not any more a cover, go to A2.

D. Use CUT to generate an inequality which cuts off the last cover. If α cuts have been generated since the last use of PRIMAL 1 or 2, go to A1; otherwise go to A2.

We call an iteration of Algorithm 2 a sequence of steps, including C2, which results either in generating a cut or in fixing some variables.

9. Numerical Example

Consider the set covering problem whose cost vector c and coefficient matrix A are shown in Tableau 3, and which is obtained from the 32-variables example of [8] by removing 9 columns (12,13,17,20,21,27,28,29,30) dominated by sums of other columns.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
c_j	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	4	4	5	5	5	8	9
A	1								1							1	1		1				1
2		1											1			1		1				1	1
3			1						1							1			1		1		1
4				1														1		1			1
5					1					1			1			1						1	
6						1													1		1		1
7							1													1		1	
8								1										1	1				1
9								1				1									1	1	
10									1											1		1	1
11											1	1		1				1	1				1
12														1	1	1			1		1	1	
13									1											1			1
14													1		1		1					1	
15									1						1				1		1	1	1

Tableau 3.

We apply Algorithm 2. To start with, we set $z_U = \sum_{j \in N} c_j = 67$,
 $z_L = 0$, $\alpha = 1$, $\beta = 2$.

Iteration 1.

A1. PRIMAL 1.

Step 0. $R_1 = \{1, \dots, 15\}$, $S_1 = \emptyset$.

Step 1. $\min_{h \in R_1} |N_h|$ is attained for $h \in I(1) = \{7, 13\}$, $N_{I(1)} = \{7, 11, 20, 22, 23\}$.

$\min_{j \in N_{I(1)}} c_j / |R_1 \cap M_j|$ is attained for $j(1) = 23$. $S_2 = \{23\}$, $R_2 = \{5, 7, 9, 12, 14\}$.

Applying Step 1 four more times, we determine the sequence of indices $j(2) = 7$, $j(3) = 8$, $j(4) = 16$, $j(5) = 5$, i.e., $S = \{5, 7, 8, 16, 23\}$.

Step 2. The vector \bar{x} defined by $\bar{x}_j = 1$, $j \in S$, $\bar{x}_j = 0$, $j \in N \setminus S$, is a prime cover. The associated upper bound is $z_U = 15$, and $T(\bar{x}) = M \setminus \{1, 2, 3, 8\}$. We set $\alpha = 2\alpha = 2$.

C1. STRENGTHEN. There is no cut yet to strengthen.

TEST 1. $Q_0 = \{j \in N \mid s_j \geq 15 - 0\} = \emptyset$ (here $s_j = c_j$, $j \in N$); we go to B1.

B1. DUAL 1.

Step 0. $R_1 = \{1, \dots, 15\}$, $u^1 = 0$, $s^1 = c$.

Step 1. $\min_{i \in R_1 \cap T(\bar{x})} |N_i| = 3$, $i(1) = \{7\}$, $u_7^2 = 1$, $u_1^2 = u_1^1$, $i \neq 7$, and

$$s^2 = (1, 1, 1, 1, 1, 1, 0, 1, 2, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5, 4, 5, 7, 9).$$

$$F_1 = \{7\}, M(1) = M_7 = \{7\}, R_2 = R_1 \setminus M(1) = \{1, \dots, 6, 8, \dots, 15\}.$$

Applying Step 1 seven more times we choose the sequence of indices $i(2) = 13$, $i(3) = 4$, $i(4) = 6$, $i(5) = 9$, $i(6) = 14$, $i(7) = 10$, $i(8) = 15$, and obtain the vectors $u^9 = (0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 2, 3, 2)$,

$s^9 = (1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 2, 0, 1, 0, 4, 0, 2, 0, 1, 0, 2)$, after which $R_9 = \emptyset$.

Step 2. $u^9 e = 12 > 0 = z_L$, hence we set $z_L = 12$ and store s^9 .

C2. TEST 2. $Q_0 = \{j \in N \mid s_j^9 \geq 15 - 12\} = \{17\}$. We set $x_{17} = 0$, $N = N \setminus \{17\}$.

Since $Q_0 \cap S(\bar{x}) = \emptyset$, and $\sum_{j \in S(\bar{x})} s_j^9 = 3 \geq z_U - u^9 e = 3$, we go to D.

D. CUT. $T(x) = M \setminus \{1, 2, 3, 8\}$, $S^+ = \{5, 23\}$, $s = s^9$.

Step 0. $W_0 = \emptyset$, $K_1 = \{5, 23\}$, $y_1 = 12$, $s^1 = s$.

Step 1. $v_1 = \min\{15-12, \max\{1, 2\}\} = 2$, $K_1^* = \{23\}$, $P_1 = \{13, 19, 23\}$,
 $M(1) = M_{23} = \{1, \dots, 4, 6, 8, 10, 11, 13, 15\}$, and

$$|N_{13} \setminus \{13, 19, 23\}| = \min_{i \in T(x) \cap M(1)} |N_i \setminus \{13, 19, 23\}| = 2.$$

Hence $i(1) = 13$, and $j(1) = N_{13} \cap K_1^* = \{23\}$. We have $W_1(13) = N_{13} \setminus P_1 = \{11, 20\}$, $y_2 = 12 + 2 = 14 < z_U = 15$, and
 $s^2 = (1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 2, 0, 1, 0, *, 0, 2, 0, 1, 0, 0)$

(here and in the following, stars replace the entries corresponding to variables that have been fixed). Further, $K_2 = \{5, 23\} \setminus \{23\} = \{5\}$.

Step 1. $v_2 = \min\{15-14, 1\} = 1$; $K_2^* = \{5\}$; $P_2 = \{1, 2, 3, 5, 10, 12, 13, 15, 19, 21\}$,
 $M(2) = \{5\}$. We have $i(2) = \{5\}$, $j(2) = \{5\}$, $W_2 = \{11, 20\} \cup \{14, 22\}$, and
 $y_3 = 14 + 1 \geq z_U = 15$.

Step 2. We add to (SC) the cut

$$x_{11} + x_{14} + x_{20} + x_{22} \geq 1$$

as the 16th inequality, and store the pairs $(W_1 = \{11, 20\}, v_1 = 2)$,
 $(W_2 \setminus W_1 = \{14, 22\}, v_2 = 1)$.

Iteration 2.

A2. PRIMAL 2.

Step 0. $R_1 = \{16\}$, $S_1 = \{5, 7, 8, 16, 23\}$

Step 1. $I(1) = \{16\}$, $N_{I(1)} = \{11, 14, 20, 22\}$, $j(1) = 11$.

$$S_2 = \{5, 7, 8, 11, 16, 23\}, R_2 = \emptyset.$$

Step 2. The vector \bar{x} whose support is $S = S_2 \setminus \{5\} = \{7, 8, 11, 16, 23\}$,
is a prime cover, with $c\bar{x} = 16$ and $T(\bar{x}) = M \setminus \{1, 2, 3, 8, 13\}$.

B2. DUAL 2.

Step 0. $m = 16$. Initialize $u^1 = (0,0,0,1,0,1,1,0,1,1,0,0,0,3,2,1,)$,

$$s^1 = (1,1,1,0,1,0,0,0,0,1,1,1,2,-1,1,0,*,0,2,1,1,-1,4), K_1^- = \{14,22\}.$$

Step 1a. $i(1) = \{14\}$, we set $u^2 = (0,0,0,1,0,1,1,0,1,1,0,0,0,2,2,1,)$,

$$s^2 = (1,1,1,0,1,0,0,0,0,1,1,1,2,0,1,1,*,1,2,1,1,0,4), K_2^- = \emptyset.$$

Step 1a. $K_2^0 = \{4,6,7,8,9,14,22\}$, $R_2 = \{3,11,13\}$.

Step 1. Since $R_2 \cap T(\bar{x}) = \{11\}$, we set $i(2) = 11$, $u_{11}^3 = \min_{j \in N_{11}} s_j^2 = 1$,

$$u_i^3 = u_i^2, i \neq 11, \text{ and } s^3 = (1,1,1,0,1,0,0,0,0,1,1,0,1,0,0,1,*,0,1,1,1,0,3).$$

Further, $F_2 = \{12,15,18\}$, $M(2) = \{2,4,8,9,11,12,14,15\}$, and $R_3 = \{3,13\}$

In two more iterations of Step 1, we set $i(3) = 13$, $u_{13}^4 = 1$, and

$i(4) = 3$, $u_3^5 = 1$, and obtain the vectors $u^5 = (0,0,1,1,0,1,1,0,1,1,1,0,1,2,2,1,)$

and $s^5 = (1,1,0,0,1,0,0,0,0,0,0,0,1,0,0,0,*,0,0,0,0,0,1)$, at which point $R_5 = \emptyset$.

Step 2. $u^5 e = 13 > 12 = z_L$, hence we set $z_L = 13$ and store s^5 in place of s^9 .

$$C2. \text{ TEST 2. } Q_0 = \{j \in N | s_j^5 \geq 15 - 13\} = \emptyset, \sum_{j \in S(\bar{x})} s_j^5 = 1 < 15 - 13,$$

hence we go to A3.

A3. PRIMAL 3.

Step 0. Initialize $S_1 = \{7,8,11,16,23\}$, $K_1 = \{1,2,5,13\}$, $R_1 = \{5,7,9,12,14,16\}$

Step 1. $\max_{j \in K_1} |M_j \cap R_1| = 1$, $j(1,1) = 5$; $j(1,2) = 11$. $S_2 = \{5,7,8,16,23\}$,

$$K_2 = \{1,2,13\}, R_2 = \{7,9,12,14,16\}.$$

Step 1. $\bigcup_{j \in K_2} M_j \cap R_2 = \emptyset$; we set $R_2 = M \setminus \bigcup_{j \in S_2} M_j = \{16\}$.

Step 2. $j(2) = \{20\}$, $S_3 = \{5,7,8,16,20,23\}$, $R_3 = \emptyset$.

Step 3. $\bigcup_{j \in S \setminus \{7\}} M_j = M$, $S = \{5,8,16,20,23\}$. Since $\sum_{j \in S(\bar{x})} s_j^5 = 2 \geq 15 - 13$, we go to D.

D. CUT generates the cut

$$x_6 + x_{11} + x_{14} + x_{19} + x_{21} + x_{22} \geq 1$$

(which becomes the 17th inequality), and the pairs $(W_1 = \{6, 19, 21\}, v_1 = 1)$, $(W_2 \setminus W_1 = \{11, 14, 22\}, v_2 = 1)$.

Since $\alpha = 2$ and 2 cuts have been generated, we go to A1.

Iteration 3.

A1. PRIMAL 1. produces the prime cover \bar{x} defined by

$S(\bar{x}) = \{4, 6, 7, 8, 10, 11, 15, 16\}$, with $T(\bar{x}) = M \setminus \{3, 12, 17\}$ and $c\bar{x} = 14$. Since $14 < z_U = 15$, we set $z_U = 15$, $\alpha \leftarrow 2\alpha = 4$, and go to C1.

C1. STRENGTHEN. Since z_U was improved by $15 - 14 = 1$, and $v_p = v_2 = 1$ for both cuts 16 and 17, each of them can be strengthened by replacing W_2 with W_1 . This yields

$$x_{11} + x_{20} \geq 1 \quad \text{and} \quad x_6 + x_{19} + x_{21} \geq 1$$

as the strengthened inequalities 16 and 17.

TEST 1. $Q_0 = \{j \in N \mid s_j^5 \geq 14 - 13\} = \{1, 2, 5, 13, 23\}$. We set $x_1 = x_2 = x_5 = x_{13} = x_{23} = 0$, $N = \{3, 4, 6, \dots, 12, 14, 15, 16, 18, \dots, 22\}$, and since \bar{x} is still a cover, we go to B1.

B1. DUAL 1 produces the vectors $u^{10} = (0, 2, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 2, 0, 2, 0, 0)$ and $s^{10} = (*, *, 1, 0, *, 0, 0, 0, 0, 1, 0, 2, *, 3, 0, 0, *, 0, 0, 0, 1, 1, *)$, with $ue = 12$.

C2. TEST 2. $Q_0 = \{j \in N \mid s_j^{10} \geq 14 - 12\} = \{12, 14\}$. We set $x_{12} = x_{14} = 0$, $N = \{3, 4, 6, \dots, 11, 15, 16, 18, \dots, 22\}$. Since $Q_0 \cap S(\bar{x}) = \emptyset$ and

$\sum_{j \in S(\bar{x})} s_j^{10} = 1 < 14 - 12$, but $\sum_{j \in N} s_j^{10} = 4 \geq 2$, we go to A3.

A3. PRIMAL 3. $S(\bar{x}) = \{4, 8, 11, 15, 16, 21, 22\}$, $c\bar{x} = 23$.

D. CUT. $x_6 + x_{19} + x_{20} \geq 1$.

Iteration 4 uses the sequence of steps A2, B2, C2, D. Step C2 sets

$x_{21} = 0$, while D generates the cut $x_{18} + x_{19} + x_{20} \geq 1$.

Iteration 5 consists of A2, B2, C2, D, and produces the cut $x_8 + x_{19} \geq 1$.

Iteration 6. A2, B1, C2, $x_{22} = 0$.

Iteration 7. A2, B1, C2, $x_9 = x_{18} = 0$.

Iteration 8. A2, B1. DUAL 1 produces the vector

$u = (0, 3, 0, 1, 2, 1, 1, 0, 1, 2, 3, 0, 0, 0, 0, 0, 0, 1, 0)$, with $u_e = 15$; we set $z_L = 15$, and since $z_L \geq z_U = 14$, the cover associated with z_U is optimal. This is \bar{x} such that $\bar{x}_j = 1$, $j = 4, 6, 7, 8, 10, 11, 15, 16$, $\bar{x}_j = 0$ otherwise.

10. Preliminary Computational Experience

A version of Algorithm 1 was implemented by Rohet Tolani in FORTRAN for the IBM 360/67 computer at CMU, and a number of test problems generated by Salkin and Koncal [9] were run. The results are summarized in Tableau 1, and compared to an involutory basis cut algorithm due to Bowman and Starr. The latter is an improvement over the Bellmore-Ratliff procedure, in that it uses a vector partial ordering to select a strong member of the family of involutory basis cuts that can be generated at each step.

Tableau 1

No.	Constraints	Variables	Iterations	Cuts	Comparison: involutory basis cuts
1	104	133	18	15	120
2	200	300	35	30	49
3	200	413	55	49	365
4	200	500	238*	237*	3000*
5	50	450	25	17	464
6	36	455	2	1	32
7	46	683	10	5	108
8	50	905	19	17	241

*Unfinished run

The discrepancy between the number of iterations and the number of cuts is due to the fact that, if some variable which is in the current cover is set to 0, the algorithm returns to step A to generate a new cover instead of generating a cut.

As can be seen from Tableau 1, the algorithm seems to converge in a surprisingly small number of iterations (and cuts), though problem 4 was still unsolved after 238 iterations, at which point the bounds were $z_U = 646$ and $z_L = 635$ (after the first iteration the upper and lower bounds were 743 and 597 respectively).

One of the surprises produced by these early runs was the considerable number of variables that could be fixed, often even in the early stages of the runs. This and the pattern of successive bound-improvements is illustrated by two typical runs shown in Tableaus 2 and 3. The numbers in column one refer to iterations where either the upper bound, or the lower bound, or the number of variables has changed. The underline means optimum value.

Tableau 2

Problem 1 (104 x 133)

Iteration	z_U	z_L	Variables left
1	1690	1637	105
2		1663	87
3	1686		
8	1682		
13	<u>1678</u>		
14		1667	76
17		1677	54
18		1717	

Tableau 3
Problem 3 (200 x 413)

Iteration	z_U	z_L	Variables left
1	827	697	413
5	785	702	391
6	759		
7	753	703	274
19		712	237
21		721	200
27	749		
41		730	161
42		731	151
45	748		
46		740	119
53		743	108
54	747		
55	<u>743</u>		

The author is currently involved in a joint effort with Andrew Ho to implement and test several versions of Algorithm 2. Apart from the merits of various heuristics, an important aspect of the project concerns the relative merits of using bit maps versus list structure. The results of this project will be reported separately.

References

- [1] E. Balas and N. Christofides, "A New Penalty Method for the Traveling Salesman Problem." Paper presented at the 9th International Symposium on Mathematical Programming, Budapest, August 1976.
- [2] E. Balas and M.W. Padberg, "Set Partitioning." B. Roy (editor), Combinatorial Programming: Methods and Applications. Reidel Publishing Co., 1975, p. 205-258.
- [3] M. Bellmore and H.D. Ratliff, "Set Covering and Involutory Bases." Management Science, 18, 1971, p. 194-206.
- [4] V.J. Bowman and J. Starr, "Set Covering by Ordinal Cuts. I: Linear Objective Functions." MSRR #321, Carnegie-Mellon University, June 1973.
- [5] N. Christofides and S. Korman, "A Computational Survey of Methods for the Set Covering Problem." Report No. 73/2, Imperial College of Science and Technology, April 1973.
- [6] R. Garfinkel and G.L. Nemhauser, "Optimal Set Covering: A Survey." A.M. Geoffrion (editor), Perspectives on Optimization. Addison-Wesley, 1972, p. 164-183.
- [7] L. B. Kovács and I. Dienes, "Maximal Direction - Complete Paths and Their Application to a Geological Problem: Setting up Stratigraphic Units." Paper presented at the 9th International Symposium in Mathematical Programming, Budapest, August 1976.
- [8] C.E. Lemke, H.M. Salkin and K. Spielberg, "Set Covering by Single Branch Enumeration with Linear Programming Subproblems." Operations Research, 19, 1971, p. 998-1022.
- [9] H.M. Salkin and R.D. Koncal, "Set Covering by an All-Integer Algorithm: Computational Experience." ACM Journal, 20, 1973, p. 189-193.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report No. 399	2. GOVT ACCESSION NO. 14 RB-399, WP-29-76-77	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Set Covering with Cutting Planes from Conditional Bounds		5. TYPE OF REPORT & PERIOD COVERED Technical Report July 1976, Rev. Jan. 1977
7. AUTHOR(s) Egon Balas		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Graduate School of Industrial Administration Carnegie-Mellon University Pittsburgh, Pennsylvania 15213		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0621, NSF-MPS MPS 73-08534 A02 NSF
11. CONTROLLING OFFICE NAME AND ADDRESS Personnel and Training Research Programs Office of Naval Research (Code 434) Arlington, Virginia 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE July 77
		13. NUMBER OF PAGES 48
		15. SECURITY CLASS. (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) 403 426 Set covering, integer programming, cutting planes		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A conditional lower bound on the minimand of an integer program is a number which would be a valid lower bound if the constraint set were amended by certain inequalities, also called conditional. If such a conditional lower bound exceeds some known upper bound, then every solution better than the one corresponding to the upper bound violates at least one of the conditional inequalities. This yields a valid disjunction, which can be used to partition the feasible set, or to derive a family of valid cutting planes. In the case of a set covering problem, these cutting planes are themselves of (over)		

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

cont

→ the set covering type. The family of valid inequalities derived from conditional bounds subsumes as a special case the Bellmore-Ratliff inequalities generated via involutory bases, but is richer than the latter class and contains considerably stronger members, where strength is measured by the number of positive coefficients. The paper discusses two algorithms based on cutting planes from conditional bounds. None of them uses the simplex method (though a variant based on the latter is also feasible). Some computational experience is presented. ↑